

EV205822397

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods And Arrangements For Providing Multiple
Concurrent Desktops And Workspaces In A Shared
Computing Environment**

Inventors:

Christopher A. Evans

Giampiero M. Sierra

Victor Tan

Praerit Garg

David Andrew Matthews

Reiner Fink

Paul S. Hellyar

RELATED APPLICATION

This application claims priority from U.S. Provisional Application Serial No. 60/170,956, filed December 15, 1999 (Applicant's Docket Number 149400.1, entitled "Multiple Desktops To Provide Separate Concurrent Work Spaces For Multiple Users Using The Same Computer", express mailing label number EL425348747US), the disclosure of which is incorporated by reference herein.

TECHNICAL FIELD

This invention relates to computers and software, and more particularly to methods and arrangements that provide a plurality of separate and concurrent desktops and workspaces within a shared computing environment.

BACKGROUND OF THE INVENTION

There are many situations where several people would like to share a single computer. For example, a home may have a single personal computer (PC) that is shared by the family members. If these users desire a unique configuration, privacy, and/or security for their applications and data files, such a shared computing environment may not adequately accommodate them.

For example, with most operating systems, the users share a single workspace. Unfortunately, doing so does not allow personalization of their workspace or security of data between users. In certain operating systems, users may separate their data into multiple "profiles" or "accounts". One of the drawbacks to this scheme is that if one user is using the computer and a another user would like to access their own files, the first user would have to close all open

1 applications in order for the other user to log on to the system and access their
2 data. Thus, the first user's application state is lost. In addition to losing the
3 application state, the user is inconvenienced because of the time involved with
4 closing down the prior user's state and re-establishing the new user's state.

5 Certain solutions, such as, for example, the "Identity Manager" available in
6 Microsoft Outlook Express version 5.0, allow certain user data to be shared among
7 separate users without losing all of the application's states during a user switch.
8 However, the Identity Manager manages only a small subset of the user's data.
9 Text documents, for example, are not stored separately for each user.
10 Furthermore, there is little workspace personalization available and the user data is
11 not secured for other user access.

12 Consequently, there is a need for an improved methods and arrangements
13 for use in multiple user computing environments. Preferably, the methods and
14 arrangements will allow for a plurality of separate and concurrent desktops and
15 workspaces within the shared computing environment

16 17 **SUMMARY OF THE INVENTION**

18 The present invention provides improved methods and arrangements for
19 use in multiple user computing environments. These methods and arrangements
20 can be configured to allow for a plurality of separate and concurrent desktops and
21 workspaces within the shared computing environment

22 The above stated needs and other are met, for example, by a method for use
23 in a multiple user computing environment logon user interface. The method
24 includes creating a separate desktop thread for each user that is authenticated
25 during a logon process, creating a separate desktop associated with each desktop

1 thread, and maintaining a list of desktop threads that are created. In this manner,
2 several users can be logged on simultaneously. In certain implementations, the
3 method further includes establishing a separate user environment associated with
4 each desktop and launching a separate user shell associated with each desktop.
5 The list of desktop threads allows for the selective switching from a first desktop
6 to a second desktop without terminating a desktop thread associated with the first
7 desktop. Indeed, in certain implementations the method includes automatically
8 switching from one desktop to another desktop without terminating the associated
9 desktop threads. For example, this can occur following a configurably defined
10 period of user inactivity. When a user logs off, then the method further includes
11 selectively removing the applicable desktop thread from the list of desktop
12 threads.

13 14 **BRIEF DESCRIPTION OF THE DRAWINGS**

15 A more complete understanding of the various methods and arrangements
16 of the present invention may be had by reference to the following detailed
17 description when taken in conjunction with the accompanying drawings wherein:

18 Fig. 1 is a block diagram depicting an exemplary computer system.

19 Fig. 2 depicts an exemplary logon screen display suitable for use in the
20 computer system of Fig. 1.

21 Fig. 3 is a flowchart depicting an exemplary process for identifying users,
22 and creating and maintaining a plurality of separate and concurrent desktops or
23 remote processes.

24 Fig. 4 is a flowchart depicting an exemplary process for switching between
25 desktops or remote processes.

DETAILED DESCRIPTION

As shown in Fig. 1, computer 20 includes one or more processors or processing units 21, a system memory 22, and a bus 23 that couples various system components including the system memory 22 to processors 21. Bus 23 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures.

The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within computer 20, such as during start-up, is stored in ROM 24.

Computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from and writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM, DVD ROM or other optical media. The hard disk drive 27, magnetic disk drive 28 and optical disk drive 30 are each connected to bus 23 by applicable interfaces 32, 33 and 34, respectively.

The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a

1 computer, such as magnetic cassettes, flash memory cards, digital video disks,
2 random access memories (RAMs) read only memories (ROM), and the like, may
3 also be used in the exemplary operating environment.

4 A number of program modules may be stored on the hard disk, magnetic
5 disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35,
6 one or more application programs 36, other program modules 37, and program
7 data 38. A user may enter commands and information into computer 20 through
8 input devices such as keyboard 40 and pointing device 42. Other input devices
9 (not shown) may include a microphone, joystick, game pad, satellite dish, scanner,
10 or the like. These and other input devices are connected to the processing unit 21
11 through an interface 46 that is coupled to bus 23.

12 A monitor 47 or other type of display device is also connected to bus 23 via
13 an interface, such as a video adapter 48. In addition to the monitor, personal
14 computers typically include other peripheral output devices (not shown) such as
15 speakers and printers.

16 Computer 20 can operate in a networked environment using logical
17 connections to one or more remote computers, such as a remote computer 50.
18 Remote computer 50 may be another personal computer, a server, a router, a
19 network PC, a peer device or other common network node, and typically includes
20 many or all of the elements described above relative to computer 20. The logical
21 connections depicted in Fig. 2 include a local area network (LAN) 51 and a wide
22 area network (WAN) 52. Such networking environments are commonplace in
23 offices, enterprise-wide computer networks, intranets, and the Internet.

24 When used in a LAN networking environment, computer 20 is connected to
25 the local network 51 through a network interface or adapter 156. When used in a

1 WAN networking environment, computer 20 typically includes a modem 54 or
2 other means for establishing communications over the wide area network 52, such
3 as the Internet. Modem 54, which may be internal or external, is connected to bus
4 23 via interface 46. In a networked environment, program modules depicted
5 relative to the personal computer 20, or portions thereof, may be stored in the
6 remote memory storage device. It will be appreciated that the network
7 connections shown are exemplary and other means of establishing a
8 communications link between the computers may be used.

9 Reference is now made to Fig. 2, which depicts an exemplary graphical
10 user interface (GUI) display suitable for use in identifying users to the operating
11 system of computer 20. Here, a logon screen 100 is shown as having a first title
12 area 102 that identifies logon screen 100. Logon screen 100 may take up be a full
13 screen of display 47 or a portion thereof. As shown, first title area 102 can include
14 any graphical feature (e.g., bitmap, photo, video, text, etc).

15 Logon screen 100 also includes a multiple user logon area 104 wherein the
16 user is presented with selectable user identifiers and related information and
17 permitted to input certain data. In this example, logon area 104 is configured for
18 five different users, namely, Billy, Christopher, Pat, Suzie, and William; each
19 being identified by a text identifier 110 and/or graphical identifier 112. Thus,
20 William may begin logging on to the operating system by selecting either text
21 identifier 110 and/or graphical identifier 112. If William has an established
22 password, then a password input field 114 is displayed and configured to receive
23 his password. Once William's password has been authenticated then William is
24 logged on. If William does not have an established password, then he would be
25 logged on after selecting text identifier 110 or graphical identifier 112. Logon

1 screen 100 can also be configured to display other user related information 116 to
2 a user. In this example, user related information 116 identifies that Suzie has ten
3 (10) new messages.

4 A second title area 106 is shown in this example along the left hand border
5 of logon screen 100. Here, second title area 106 identifies the computer as the
6 "Den Computer". A selectable shut down mechanism 108 is also provided to
7 allow a user to shut down the computer.

8 Logon screen 100 is, by way of example, the starting point for this
9 description of the various methods and arrangements that allow for a plurality of
10 separate and concurrent desktops and workspaces within the shared computing
11 environment of computer 20. As described in more detail below and depicted in
12 the flowcharts of Figs.3-4, a desktop switching scheme is implemented for use
13 with Microsoft Windows™ NT or Windows™ 2000. Those skilled in the art,
14 however, will recognize that the desktop switching scheme is applicable to other
15 operating systems, computers, etc. For example, the scheme is extended herein to
16 further include WinStation switching, each of which may include desktop
17 switching as well.

18 The exemplary implementations presented herein utilize the Desktop
19 application programming interfaces (APIs) available in Windows NT to create
20 separate workspaces for each user who interactively logs on through logon screen
21 100. Using Windows NT profiles, the user's data is separated from other user's
22 data and depending on the user's or the computer manager's preferences, access to
23 other user's data can be secured. By using a separate desktop for each user,
24 multiple users can be logged on to the computer simultaneously. Switching users
25 is very fast, since it is just a desktop switch. No application state is lost during the

1 switch because the applications run within the context of a desktop and are
2 completely separated from one another.

3 To protect the user's logon session from other users if the users should
4 happen to walk away from the computer without securing their workstation, the
5 operating system will automatically switch back to the logon desktop after a user
6 configurable timeout period. The logon desktop is secured from the user desktops
7 and can only switch back to a user desktop if the user is identified by an
8 authentication entity within the operating system.

9 With this in mind, attention is drawn to the flowchart in Fig. 3. Here,
10 process 200 begins with the computer displaying logon screen 100. In step 202,
11 the user provides the necessary inputs to logon as part of the user authentication
12 scheme. For example, the user selects text identifier 110 or graphical identifier
13 112 and subsequently enters a password in password input field 114. The
14 authentication process occurs in the operating system, as depicted by decision step
15 204. If the authentication was not a success, then in accord with step 204 (NO),
16 process 200 returns to logon screen 100. If the authentication was a success, then
17 in accord with step 204 (YES), process 200 continues to decision step 206.

18 In decision step 206, it is determined if the authenticated user from step 204
19 is already logged in. This is accomplished by checking open desktops list (ODL)
20 208. If the authenticated user has not already logged in per step 206 (NO), then in
21 accordance with step 210 a new desktop thread is created for this user. In step
22 214, the thread and desktop information are then added to the list of users
23 currently logged on within ODL 208. The user's thread also sets up the user
24 environment in step 216, and starts the user shell on their newly created desktop in
25

1 step 218. In step 220 the logon program switches to the user's new desktop 222.

2 At this point, the logon process is complete.

3 Returning to decision step 206, if a desktop is already open for the user as
4 demonstrated by an associated entry in ODL 208, then step 206 (YES) advances
5 process 200 to step 224. In step 224, a switch is made to the existing desktop
6 thread, followed by a switch to the applicable desktop 222, in step 220. In Fig. 3
7 there are $N > 1$ desktops as represented by desktops 222a-c.

8 Reference is now made to Fig. 4, wherein a flowchart depicts a process 240
9 for logging off of a desktop 222. In this example, there are three ways for the user
10 to become "logged off". The first takes place in step 242, wherein the user
11 intentionally and fully logs off. When such a logoff event is generated in step 242,
12 then in step 244 the desktop is cleaned up, ODL 208 is updated and the thread is
13 terminated. After step 244, process 240 returns to logon screen 100.

14 The second way for a user to become "logged off" is provided in step 246,
15 wherein the user switches desktops manually. At any point prior to logging off,
16 the user can initiate an action (e.g., by user input), which will return them to logon
17 screen 100 without signaling the user's thread that a logoff has occurred. When
18 this happens, the users desktop remains running and all applications that have been
19 started on that desktop continue to run. However, the user is unable to see any of
20 these applications because their desktop is essentially hidden. At this point, the
21 desktop is said to be "switched out", but it is still active. The user will need to log
22 on again in accord with process 200 to have the active desktop "switched in".

23 The third way that a user can become "logged off" is shown in step 248,
24 wherein after a user configurable period of inactivity (e.g., no mouse activity or
25 keyboard input) the operating system will assume that the user has stopped

1 working on the computer for the time being and it will automatically switch back
2 to logon screen 100. This helps to improve security by securing the user's desktop
3 when they no longer appear to be using the computer. It also makes it easier for
4 other users to utilize the computer and not worry about ending another user's
5 session.

6 It is also possible to provide a plurality of separate and concurrent
7 environments within the shared computing environment by switching remote
8 processes, such as, for example, WinStations, instead of desktops. Logically,
9 WinStations contain multiple desktops. Creating a WinStation for each user and
10 switching the active WinStation when a user logs on or switches back to logon
11 screen 100 could implement the same mechanism as described above.

12 Although some preferred embodiments of the various methods and
13 arrangements of the present invention have been illustrated in the accompanying
14 Drawings and described in the foregoing Detailed Description, it will be
15 understood that the invention is not limited to the exemplary embodiments
16 disclosed, but is capable of numerous rearrangements, modifications and
17 substitutions without departing from the spirit of the invention as set forth and
18 defined by the following claims.